(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号
特開2003－15522
（P2003－15522A）
(43)公開日　平成15年１月17日(2003.1.17)

(51)Int.Cl.'	識別記号	FI	テーマコード'(参考)
G09C 1/00	610	G09C 1/00	610A 5J104

審査請求　未請求　請求項の数５　ＯＬ　（全 22 頁）

(21)出願番号　特願2001－195752(P2001－195752)

(22)出願日　平成13年６月28日(2001.6.28)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　岡田　壮一
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

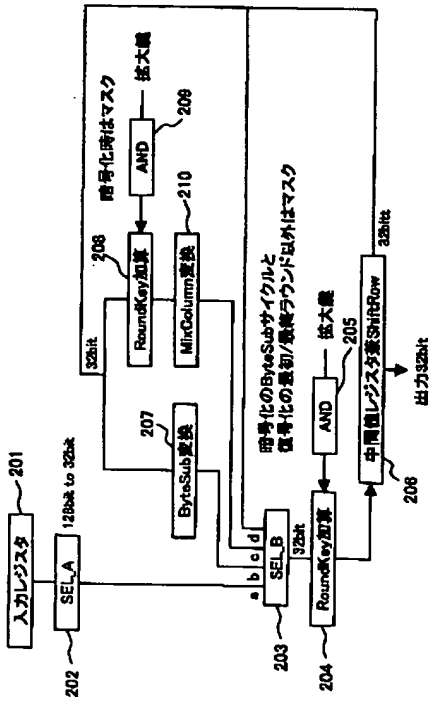
(72)発明者　鳥居　直哉
神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内

(74)代理人　100094145
弁理士　小野　由己男　（外２名）

最終頁に続く

(54)【発明の名称】　暗号回路

(57)【要約】
【課題】　AESブロック暗号を実装する際に、回路規模を小型化するとともに一定水準の高速処理が実現可能な暗号回路を提供する。
【解決手段】　Round処理部が、Round Keyの値を入力データに加算する第１Round Key加算回路と、第１Round Key加算回路の出力を一時的に格納するとともにShift Row変換を実行する中間レジスタ兼Shift Row変換回路と、中間レジスタ兼Shift Row変換回路の値が入力されByte Sub変換を実行するByte Sub変換回路と、中間レジスタ兼Shift Row変換回路の値が入力されRound Keyの値を加算する第２Round Key加算回路と、第２Round Key加算回路の出力に対してMix Column変換を実行するMix Column変換回路と、第１セレクタ、中間レジスタ兼Shift Row変換回路、Byte Sub変換回路、Mix Column変換回路の出力のうちいずれか１つを第２Round Key加算回路に出力する第２セレクタとを備える。



1

【特許請求の範囲】

【請求項１】暗号鍵から所定の処理ブロック長に対応するビット数の複数のRound Keyを生成し、入力データと前記Round Keyのうちの１つと排他的論理和演算を実行するEXOR演算部と、Byte Sub変換、Shift Row変換、Mix Column変換、Round Key加算を含むRound処理を複数回実行するRound処理部とを備えるラウンドファンクション部により入力データと前記Round Keyによる暗号化／復号化処理を前記処理ブロック長毎に実行する暗号回路であって、

前記Round処理部が、入力データを前記処理ブロック長よりも小さい実行ブロック長に分割する第１セレクタと、前記実行ブロック長毎に前記Round Keyの値を入力データに加算する第１Round Key加算回路と、前記第１Round Key加算回路の出力を一時的に格納するとともに前記処理ブロック長によるShift Row変換を実行する中間レジスタ兼Shift Row変換回路と、前記中間レジスタ兼Shift Row変換回路の値が前記実行ブロック長毎に入力されByte Sub変換を実行するByte Sub変換回路と、前記中間レジスタ兼Shift Row変換回路の値が前記実行ブロック長毎に入力され前記Round Keyの値を前記実行ブロック長毎に加算する第２Round Key加算回路と、前記第２Round Key加算回路の出力に対してMix Column変換を実行するMix Column変換回路と、前記第１セレクタ、中間レジスタ兼Shift Row変換回路、Byte Sub変換回路、Mix Column変換回路の出力のうちいずれか１つを前記第２Round Key加算回路に出力する第２セレクタとを備えることを特徴とする暗号回路。

【請求項２】前記Byte Sub変換回路は、入力データに対して行列演算を実行する復号化用行列演算部と、入力データと前記復号化用行列演算部の出力とのうちいずれかを出力する第３セレクタと、前記第３セレクタから出力されるデータに対して逆数演算を実行する逆数演算部と、前記逆数演算部から出力されるデータに対して行列演算を実行する暗号化用行列演算部と、前記逆数演算部の出力と前記暗号化用行列演算部の出力のうちいずれか一方を出力する第４セレクタとを備える請求項１に記載の暗号回路。

【請求項３】前記中間レジスタ兼Shift Row変換回路に入力されるデータのシフト量に関するシフトデータを暗号化時と復号化時において逆順に入力することにより、前記中間レジスタ兼Shift Row変換回路を暗号化と復号化に共通に使用する、請求項１または２に記載の暗号回路。

【請求項４】前記Mix Column変換回路は、乗数固定の複数の乗算器と前記複数の乗算器の排他的論理和を演算するEXOR回路とを備え、各乗算器に入力されるデータと各乗算器に設定された乗数との間で行列演算を実行する、請求項１～３のいずれかに記載の暗号回路。

【請求項５】暗号鍵を前記実行ブロック長に応じたビット数に分割して出力する第５セレクタと、

2

前記実行ブロック長毎にデータをラッチするフリップフロップ回路が複数段接続されたシフトレジスタと、

前記シフトレジスタの最終段のフリップフロップ回路の出力と定数群のうちから選択される１つの定数との排他的論理和演算を実行する第１EXOR回路と、

前記シフトレジスタのフリップフロップのうち暗号化時に演算対象となるフリップフロップの出力と復号化時に演算対象となるフリップフロップの出力とが入力されいずれか１つを選択的に出力する第６セレクタと、

前記第６セレクタの出力をローテーション処理するRot Byte処理回路と、

前記第６セレクタの出力と前記Rot Byte処理回路の出力とが入力され、いずれか１つを選択的に出力する第７セレクタと、

前記第７セレクタの出力に対して前記実行ブロック長毎のByte Sub変換を実行するSub Byte処理回路と、

前記第６セレクタの出力と前記Sub Byte処理回路の出力が入力され、いずれか１つを選択的に出力する第８セレクタと、

前記第１EXOR回路の出力と前記第８セレクタの出力とに基づいて排他的論理和演算を実行する第２EXOR回路と、

前記シフトレジスタのフリップフロップのうち暗号化時にその出力が演算対象となるフリップフロップに対して、前記第２EXOR回路の出力と、隣接する段のフリップフロップの出力とのうちいずれかを選択的に出力するシフトレジスタ部セレクタと、を備える、前記暗号鍵から所定の処理ブロック長に対応するビット数の複数のRound Keyを前記実行ブロック長に対応するビット数毎に分割された拡大鍵として生成する拡大鍵スケジュール回路を有する請求項１～４のいずれかに記載の暗号回路。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、共通鍵暗号として現在の米国標準であるDESに代わり、次世代の共通鍵ブロック暗号AES (Advanced Encryption Standard) として標準化されつつあるRijndaelアルゴリズムをハードウェア実装するための暗号回路に関する。

【０００２】

【従来の技術】電子商取引や電子マネー等、インターネットを用いた様々なサービスが検討されている。これらの技術は、個人の生活にとどまらず、会社間の取引や生産性の向上等様々な分野で広く用いられていく技術である。特に、今後は、個人認証を目的として、スマートカードやモバイル機に暗号機能が搭載され、認証、電子署名、及びデータの暗号化等に広く使われると予想される。

【０００３】これらの応用において、インターネット上の第三者からの盗聴を防ぐ意味で共通鍵暗号が用いられる。共通鍵暗号は、現在の米国標準であるDESに代わる

次世代の共通鍵ブロック暗号AES（Advanced Encryption Standard）として、Rijndaelと呼ばれる暗号アルゴリズムが選定され、標準化されようとしている（AESドラフト http://csrc.nist.gov/publications/drafts/dfips-AES.pdf参照）。

【０００４】AESは処理ブロック長128ビットのブロック暗号であり、その暗号化アルゴリズムは、図１に示すように、ラウンドファンクション部２０と鍵スケジュール部１０で構成される暗号回路により実現可能であると考えられる。ラウンドファンクション部２０は、入力データを一時格納する入力レジスタ２１と、入力データと拡大鍵との排他的論理和演算を行うEXOR処理部２２とRound処理部２３、最終Round処理部２４および出力データを一時格納する出力レジスタ２５からなる。

【０００５】Round処理部２３は、Byte Sub変換部３１、Shift Row変換部３２、Mix Column変換部３３およびRound Key加算部３４からなり、最終Round処理部２４はRound処理部２３からMix Column変換部３３を除いた処理を行うもので、Byte Sub変換部３５、 Shift Row変換部３６およびRound Key加算部３７から構成される。

【０００６】Round処理は繰り返し行われるが、最終Round処理を含むRound処理の繰り返し回数Nrは、鍵スケジュール部１０に入力される鍵長により決められており、表１に示すように規定されている。

【０００７】

【表１】

鍵長とラウンド数

鍵長	Nr
128bit	10
192bit	12
256bit	14

【０００８】このことから、各鍵長においてそれぞれRound処理の回数をNr-1回行い、最後に最終Round処理を行うものであり、鍵長が128bitではRound処理が9回繰り返され、鍵長が192bitでは11回繰り返され、鍵長が256bitでは13回繰り返され、最後に最終Round処理を行うこととなる。EXOR回路２２、Round処理部２３および最終Round処理部２４には、鍵スケジュール部１０で生成されたRound Keyが入力される。

【０００９】鍵スケジュール部１０は、 AESドラフトに示される鍵生成スケジュールに基づいてRound Keyを生成するものであり、そのアルゴリズムを図２に示す。AESブロック暗号の回路化にあたり、AES Proposal仕様（Rijndael Specification, http://csrc.nist.gov/encryption/aes/rijndael/Rijndael.pdf）に２つのハードウェア実装が紹介されている。

【００１０】そのうちの１つは、図１に示す全ての機能をそのまま128bit単位でハードウェア化する方法（以下、従来例１と称す）である。この場合、暗号化と復号化において、各機能の処理の順番が逆となり、暗号化における処理回路と復号化時における処理回路を別々に用意する必要がある。

【００１１】また、表１に示すように鍵長によりRound処理の実施回数を変える必要があるため、鍵長毎に回路を作成する必要がある。さらに、ラウンドファンクション部２０で使用されるRound Keyは、暗号化時と復号化時において逆順になるため、鍵スケジュール部１０において鍵生成の順を暗号化時と復号化時とで逆にする必要がある。したがって、暗号化時と復号化時における鍵スケジュール回路を別々に構成するか、暗号化時と復号化時において鍵スケジュール部１０を共通に使用できるような工夫をする必要がある。

【００１２】２つ目の方法は、図３に示すように、Byte Sub変換部５１とMix Column変換部５２を有するコプロセッサ５０を作成してByte Sub変換とMix Column変換の機能のみをハードウェア化し、その他の機能をプログラム４１に組み込んでソフトウェア化しＣＰＵ４０で処理する方法（以下、従来例２と称す）である。

【００１３】この場合、処理時間的にＣＰＵ４０での処理に不向きなByte Sub変換とMix Column変換をコプロセッサ５０としてハードウェア化し、他の処理をＣＰＵ４０に格納されたプログラム４１で処理することで回路規模を小さくすることができる。

【００１４】

【発明が解決しようとする課題】スマートカード等にAESブロック暗号を組込むことを想定した場合、暗号回路に要求される性能は、一定水準の処理速度を保ちつつ、回路規模の小型化にある。 この要求を前提とした場合、従来提案されている全ての機能を128bit単位でハードウェア化する方法では回路規模が大きすぎ、スマートカードへの搭載が困難となる。また、Byte Sub変換とMix Column変換の機能のみをハードウェア化し、他機能をソフトウェアで処理する方法では、要求処理速度を満足しないといった問題がある。

【００１５】また、Round Keyを生成する鍵スケジュール部１０では、メモリにすべてのRound Keyを格納すると大きなメモリ容量が必要となり、回路規模が大きくなる。したがって、処理速度を低下させずに回路規模の削減を行うために、メモリにすべての拡大鍵を格納する必要がないような回路構成でRound Keyの生成を行うことが望まれる。

【００１６】本発明では、AESブロック暗号を実装する際に、回路規模を小型化するとともに一定水準の高速処理が実現可能な暗号回路の提供を目的とする。

【００１７】

【課題を解決するための手段】本発明に係る暗号回路は、暗号鍵から所定の処理ブロック長に対応するビット数の複数のRound Keyを生成し、入力データとRound Keyのうちの１つとEXOR演算を実行するEXOR演算部と、Byte

Sub変換、Shift Row変換、Mix Column変換、Round Key加算を含むRound処理を複数回実行するRound処理部とを備えるラウンドファンクション部により入力データとRound Keyによる暗号化／復号化処理を処理ブロック長毎に実行する暗号回路であって、Round処理部が、入力データを処理ブロック長よりも小さい実行ブロック長に分割する第１セレクタと、実行ブロック長毎にRound Keyの値を入力データに加算する第１Round Key加算回路と、第１Round Key加算回路の出力を一時的に格納するとともに処理ブロック長によるShift Row変換を実行する中間レジスタ兼Shift Row変換回路と、中間レジスタ兼Shift Row変換回路の値が実行ブロック長毎に入力されByte Sub変換を実行するByteSub変換回路と、中間レジスタ兼Shift Row変換回路の値が実行ブロック長毎に入力されRound Keyの値を実行ブロック長毎に加算する第２Round Key加算回路と、第２Round Key加算回路の出力に対してMix Column変換を実行するMix Column変換回路と、第１セレクタ、中間レジスタ兼Shift Row変換回路、Byte Sub変換回路、Mix Column変換回路の出力のうちいずれか１つを第２Round Key加算回路に出力する第２セレクタとを備える。

【００１８】ここで、実行ブロック長は８の倍数ビットとすることができ、処理ブロック長が１２８ビットであり実行ブロック長が３２ビットとすることができる。さらに、暗号鍵の鍵長が１２８ビット、１９２ビット、２５６ビットのうちのいずれかとすることができる。

【００１９】また、Byte Sub変換回路は、入力データに対して行列演算を実行する復号化用行列演算部と、入力データと復号化用行列演算部の出力とのうちいずれかを出力する第３セレクタと、第３セレクタから出力されるデータに対して逆数演算を実行する逆数演算部と、逆数演算部から出力されるデータに対して行列演算を実行する暗号化用行列演算部と、逆数演算部の出力と暗号化用行列演算部の出力のうちいずれか一方を出力する第４セレクタとを備える構成とすることができる。

【００２０】さらに、復号化用行列演算部と暗号化行列演算部は、１クロックで８ビット毎の演算を行うように複数の排他的論理和演算回路を備えている構成とすることができ、１クロックで１ビットの演算を行うように排他的論理和演算回路を備える構成とすることができる。

【００２１】また、中間レジスタ兼Shift Row変換回路に入力されるデータのシフト量に関するシフトデータを暗号化時と復号化時において逆順に入力することにより、中間レジスタ兼Shift Row変換回路を暗号化と復号化に共通に使用することができる。

【００２２】さらに、Mix Column変換回路は、乗数固定の複数の乗算器と複数の乗算器の排他的論理和を演算するEXOR回路とを備え、各乗算器に入力されるデータと各乗算器に設定された乗数との間で行列演算を実行するように構成できる。この場合、Mix Column変換回路は、８ビット単位で演算が可能な４つの乗算器と４つの乗算器の出力に基づいて排他的論理和演算を実行するEXOR回路とを有する演算器を４つ備える構成とすることができる。この乗算器は、２つの乗数を制御可能であり暗号化と復号化に共通に使用されるように構成でき、上位ビットからの加算値を制御するように構成することもできる。

【００２３】また、暗号鍵を実行ブロック長に応じたビット数に分割して出力する第５セレクタと、実行ブロック長毎にデータをラッチするフリップフロップ回路が複数段接続されたシフトレジスタと、シフトレジスタの最終段のフリップフロップ回路の出力と定数群のうちから選択される１つの定数との排他的論理和演算を実行する第１EXOR回路と、シフトレジスタのフリップフロップのうち暗号化時に演算対象となるフリップフロップの出力と復号化時に演算対象となるフリップフロップの出力とが入力されいずれか１つを選択的に出力する第６セレクタと、第６セレクタの出力をローテーション処理するRot Byte処理回路と、第６セレクタの出力とRot Byte処理回路の出力とが入力され、いずれか１つを選択的に出力する第７セレクタと、第７セレクタの出力に対して実行ブロック長毎のByte Sub変換を実行するSub Byte処理回路と、第６セレクタの出力とSub Byte処理回路の出力が入力され、いずれか１つを選択的に出力する第８セレクタと、第１EXOR回路の出力と第８セレクタの出力とに基づいて排他的論理和演算を実行する第２EXOR回路と、シフトレジスタのフリップフロップのうち暗号化時にその出力が演算対象となるフリップフロップに対して、第２EXOR回路の出力と、隣接する段のフリップフロップの出力とのうちいずれかを選択的に出力するシフトレジスタ部セレクタとを備える、暗号鍵から所定の処理ブロック長に対応するビット数の複数のRoundKeyを実行ブロック長に対応するビット数毎に分割された拡大鍵として生成する拡大鍵スケジュール回路を有するように構成できる。

【００２４】ここで、シフトレジスタは、３２ビット単位でデータ処理を実行する８個のフリップフロップ回路を備え、第６セレクタは、フリップフロップのうち下から２段目、４段目、６段目、８段目のフリップフロップの出力が入力され、いずれか１つを出力するように構成できる。

【００２５】また、第７セレクタに中間レジスタ兼Shift Row変換回路の出力が入力され、Sub Byte処理回路の出力を第２セレクタに入力することにより、Sub Byte処理回路とRound処理部のByte Sub変換回路とを共用するように構成できる。

【００２６】

【発明の実施の形態】〈ラウンドファンクション部〉AESブロック暗号は、鍵長１２８／１９２／２５６ビット、処理ブロック長１２８ビットで暗号化／復号化を行

うものであり、図１に示すように、暗号鍵から複数のRound Keyを生成する鍵スケジュール部１０と、鍵スケジュール部１０から供給されるRound Keyを用いて暗号化／復号化を実行するラウンドファンクション部２０で構成される。ラウンドファンクション部２０では、排他的論理和演算、Byte Sub変換処理、Shift Row変換処理、Mix Column変換処理、Round Key加算処理などの処理を行う。

【００２７】第１実施形態は、このラウンドファンクション部２０を実現するための回路を構成するものであり、図４にその回路構成を示す。各回路ブロックではShift Row変換処理を除いて３２ビットによる処理を実行し、Shift Row変換処理では１２８ビットによる処理を実行するものとし、各回路ブロック間のデータの転送は３２ビット単位で実施されるものとする。

【００２８】このラウンドファンクション部には、入力データを一時格納する入力レジスタ２０１と、１２８ビットの入力データから３２ビットのデータを選択する第１セレクタ２０２と、第１セレクタ２０２の出力が１つの入力端子に入力される第２セレクタ２０３と、第２セレクタ２０３の出力が入力される第１Round Key加算回路２０４と、第１Round Key加算回路２０４に拡大鍵または"0"を加算データとして入力する加算データセレクタ２０５と、第１Round Key加算回路２０４の出力値を格納するとともに１２８ビット単位でShift Row変換処理を実行する中間レジスタ兼Shift Row変換回路２０６と、中間レジスタ兼Shift Row変換回路２０６の値が３２ビット毎に入力されByte Sub変換を実行するByte Sub変換回路２０７と、中間レジスタ兼Shift Row変換回路２０６の値が３２ビット毎に入力される第２Round Key加算回路２０８と、第２Round Key加算回路２０８に拡大鍵または"0"を加算データとして入力する加算データセレクタ２０９と、第２Round Key加算回路２０８の出力に対してMix Column変換を実行するMix Column変換回路２１０とを備えている。第２セレクタ２０３には、第１セレクタ２０２、Byte Sub変換回路２０７、Mix Column変換回路２１０および中間レジスタ兼Shift Row変換回路２０６の出力が入力されており、このうちいずれか１つを第２Round Key加算回路２０４に出力するように構成される。

〈暗号化時の動作スケジュール〉このラウンドファンクション部における暗号化時における動作スケジュールを表２に示す。

【００２９】

【表２】

ラウンドファンクション動作スケジュール

Round	Cycle	処理	SEL_B
0	000-003	RounKey加算	a
1	004-007	ByteSub変換	b
	008	ShiftRow変換	c
	009-012	MixColumn変換 RoundKey加算	c
2	013-016	ByteSub変換	b
	017	ShiftRow変換	c
	018-021	MixColumn変換 RoundKey加算	c
	省略		
Nr-1	#1	ByteSub変換	b
	(Nr-1)*9-1	ShiftRow変換	c
	(Nr-1)*9 – (Nr-1)*9+3	MixColumn変換 RoundKey加算	c
Nr	#2	ByteSub変換	b
	Nr*9-1	ShiftRow変換	d
	Nr*9 – Nr*9+3	RoundKey加算	d

#1:(Nr-1)*9-5 – (Nr-1)*9-2
#2:Nr*9-5 – Nr*9-2

注:表は暗号化の動作を示している
復号化ではRound KeyとMixColumnの
処理順番が入れ代る

【００３０】ここで、Round０では、第２セレクタ２０３のセレクタ位置をａとして第１Round Key加算回路２０４により拡大鍵の加算処理を実行する。第１セレクタ２０２により入力レジスタ２０１内の入力データを３２ビット毎に選択して第１RoundKey加算回路２０４に入力し、鍵スケジュール部から入力されるRound Keyのうち３２ビット毎に分割された拡大鍵の加算処理を実行する。入力データおよび拡大鍵を３２ビット毎に変更しながら、第１Round Key加算回路２０４による加算処理を実行することで、第000サイクル～第003サイクルの４サイクルで１２８ビットの処理ブロックに対し、図１におけるEXOR処理部２２の排他的論理和演算処理を実行することができる。第１Round Key加算回路２０４による演算結果は、３２ビット毎に中間レジスタ兼Shift Row変換回路２０６に順次格納される。

【００３１】Round１では、図１におけるRound処理２３を実行するものであり、Byte Sub変換処理３１、Shift Row変換処理３２、Mix Column変換処理３３、Round Key加算処理３４を実行する。このために、まず、第004サイクル～第007サイクルで、第２セレクタ２０３のセレクタ位置をｂとし、中間レジスタ兼Shift Row変換回路２０６に格納されているデータを３２ビットずつシフトさせながら読み出してByte Sub変換回路２０７に入力する。このとき、加算データセレクタ２０５により選択さ

れるデータを"0"とすることで、第１Round Key加算回路２０４をマスク状態とする。Byte Sub変換回路２０７による演算結果は、３２ビット毎に中間レジスタ兼Shift Row変換回路２０６に順次格納される。このことにより、１２８ビットのByte Sub変換処理が行われ、演算結果が中間レジスタ兼Shift Row変換回路２０６に格納されることとなる。

【００３２】次に、第008サイクルにおいて、Shift Row変換処理を実行する。中間レジスタ兼Shift Row変換回路２０６では、１２８ビット単位でShift Row変換処理を行うことが可能となっており、この第008サイクルにおいて、１２８ビットのShift Row変換処理を実行する。このとき、第２セレクタ２０３のセレクタ位置はいずれの位置であってもよいが、次のサイクルでの処理を考慮してｃの位置とすることが好ましい。

【００３３】第009サイクル～第012サイクルでは、Mix Column変換処理およびRound Key加算処理を実行する。ここでは、中間レジスタ兼Shift Row変換回路２０６に格納されているデータを３２ビットずつシフトさせながら読み出して第２Round Key加算回路２０８に入力する。このとき、加算データセレクタ２０９により選択されるデータを"0"とすることで、第２Round Key加算回路２０８をマスク状態とする。また、第２セレクタ２０３のセレクタ位置をｃとすることにより、Mix Column変換回路２１０でMix Column変換処理が実行されたデータが第２セレクタ２０３を介して第１Round Key加算回路２０４に入力される。加算データセレクタ２０５により選択されるデータは、鍵スケジュール部から入力される拡大鍵が選択されており、第１Round Key加算回路２０４においてRound Key加算処理が実行される。Mix Column変換回路２１０でMix Column変換処理され、第１Round Key加算回路２０４においてRound Key加算処理された演算結果は、３２ビット毎に中間レジスタ兼Shift Row変換回路２０６にそれぞれシフトされながら格納される。このことにより、第009サイクル～第012サイクルで、Mix Column変換処理およびRound Key加算処理を実行した１２８ビットの演算結果が中間レジスタ兼ShiftRow変換回路２０６に格納される。このようにして、第004サイクル～第012サイクルの９サイクルで１つのRound処理が実行される。

【００３４】以下、第２Round～第（Nr－１）Roundにおいて、第１Roundと同様の処理を実行する（ただし、Nrは最終Round処理を含むRound処理の繰り返し回数であり、表１に示すように、鍵長により異なる回数が規定されている）。

【００３５】第（Nr）Round（最終Round）では、図１における最終Round処理２４を実行するものであり、Byte Sub変換処理３５、Shift Row変換処理３６、Round Key加算処理３７を実行する。

【００３６】このために、第（Nr＊９－５）サイクル～第（Nr＊９－２）サイクルにおいて、第２セレクタ２０３のセレクタ位置をｂとし、中間レジスタ兼Shift Row変換回路２０６に格納されているデータを３２ビットずつシフトさせながら読み出してByte Sub変換回路２０７に入力する。このとき、加算データセレクタ２０５により選択されるデータを"0"とすることで、第１Round Key加算回路２０４をマスク状態とする。Byte Sub変換回路２０７による演算結果は、３２ビット毎に中間レジスタ兼Shift Row変換回路２０６に順次格納される。このことにより、１２８ビットのByte Sub変換処理が行われ、演算結果が中間レジスタ兼Shift Row変換回路２０６に格納されることとなる。

【００３７】次に、第（Nr＊９－１）サイクルにおいて、１２８ビットのShift Row変換処理を実行する。このとき、第２セレクタ２０３のセレクタ位置はいずれの位置であってもよいが、次のサイクルでの処理を考慮してｄの位置とすることが好ましい。

【００３８】第（Nr＊９）サイクル～第（Nr＊９＋３）サイクルでは、Round Key加算処理を実行する。ここでは、第２セレクタ２０４のセレクタ位置をｄとすることで、中間レジスタ兼Shift Row変換回路２０６に格納されているデータを３２ビットずつシフトさせながら読み出して第２セレクタ２０３を介して第１Round Key加算回路２０４に入力する。このとき、加算データセレクタ２０５により選択されるデータを鍵スケジュール部から供給される拡大鍵とすることで、第１Round Key加算回路２０４による３２ビット毎のRound Key加算処理を実行することができる。第１Round Key加算回路２０４においてRound Key加算処理された演算結果は、３２ビット毎に中間レジスタ兼Shift Row変換回路２０６にそれぞれシフトされながら格納される。このことにより、第（Nr＊９）サイクル～第（Nr＊９＋３）サイクルで、Round Key加算処理を実行した１２８ビットの演算結果が中間レジスタ兼Shift Row変換回路２０６に格納される。このようにして、第（Nr＊９－５）サイクル～第（Nr＊９＋３）サイクルの９サイクルで最終Round処理が実行される。

〈復号化時の動作スケジュール〉このラウンドファンクション部における復号化時の動作は暗号化時の逆順で処理が実行される。このときの動作スケジュールを表３に示す。

【００３９】

【表３】

11

ラウンドファンクション動作スケジュール

Round	Cycle	処理	SEL_B
0	000-003	RounKey加算	a
1	004	ShiftRow変換	b
	005-008	ByteSub変換	b
	009-012	RounKey加算 MixColumn変換	c
2	013	ShiftRow変換	b
	014-017	ByteSub変換	b
	018-021	RounKey加算 MixColumn変換	a
	省略		
Nr-1	(Nr-1)*9-5	ShiftRow変換	b
	#1	ByteSub変換	b
	(Nr-1)*9 – (Nr-1)*9+3	RounKey加算 MixColumn変換	a
Nr	Nr*9-5	ShiftRow変換	b
	#2	ByteSub変換	b
	Nr*9 – Nr*9+3	RoundKey加算	d

#1:(Nr-1)*9-4 – (Nr-1)*9-1
#2:Nr*9-4 – Nr*9-1

【００４０】ここで、Round０では、第２セレクタ２０３のセレクタ位置をａとして第１Round Key加算回路２０４により拡大鍵の加算処理を実行する。第１セレクタ２０２により入力レジスタ２０１内の入力データを３２ビット毎に選択して第１RoundKey加算回路２０４に入力し、鍵スケジュール部から入力されるRound Keyのうち３２ビット毎に分割された拡大鍵の加算処理を実行する。このとき、第１セレクタ２０２を介して入力されるデータは暗号化時の入力順と逆に入力され、かつ鍵スケジュール部から入力される拡大鍵の入力順も暗号化時の逆に入力することとする。このように、入力データおよび拡大鍵を３２ビット毎に変更しながら、第１Round Key加算回路２０４による加算処理を実行することで、第000サイクル～第003サイクルの４サイクルで１２８ビットの処理ブロックに対するRound Key加算処理を実行することができる。第１Round Key加算回路２０４による演算結果は、３２ビット毎に中間レジスタ兼Shift Row変換回路２０６に順次格納される。

【００４１】Round１では、Shift Row変換、Byte Sub変換、Round Key加算、Mix Column変換の順に処理を実行する。このために、まず、第004サイクルにおいて、中間レジスタ兼Shift Row変換回路２０６において、１２８ビット単位でShift Row変換処理を実行する。この場合、暗号化時におけるShift Row変換処理と同じ処理が行われる。また、第２セレクタ２０３のセレクタ位置はいずれの位置であってもよいが、次のサイクルでの処理を考慮してｂの位置とすることが好ましい。

12

【００４２】次に、第005サイクル～第008サイクルで、第２セレクタ２０３のセレクタ位置をｂとし、中間レジスタ兼Shift Row変換回路２０６に格納されているデータを３２ビットずつシフトさせながら読み出してByte Sub変換回路２０７に入力する。このとき、加算データセレクタ２０５により選択されるデータを"0"とすることで、第１Round Key加算回路２０４をマスク状態とする。Byte Sub変換回路２０７による演算結果は、３２ビット毎に中間レジスタ兼Shift Row変換回路２０６に順次格納される。この場合も、暗号化時の逆変換処理となるようにByte Sub変換処理を実行するものであり、詳細は後述する。このことにより、１２８ビットのByte Sub変換処理が行われ、演算結果が中間レジスタ兼Shift Row変換回路２０６に格納されることとなる。

【００４３】第009サイクル～第012サイクルでは、Round Key加算処理およびMix Column変換処理を実行する。ここでは、中間レジスタ兼Shift Row変換回路２０６に格納されているデータを３２ビットずつシフトさせながら読み出して第２Round Key加算回路２０８に入力する。このとき、加算データセレクタ２０９により選択されるデータを鍵スケジュール部から入力される拡大鍵とする。また、第２セレクタ２０３のセレクタ位置をｃとして、Mix Column変換回路２１０の出力を第２セレクタ２０３を介して第１Round Key加算回路２０４に入力する。このとき、加算データセレクタ２０５により選択されるデータを"0"とすることにより、第１Round Key加算回路２０４をマスク状態とする。この場合も、暗号化時と逆変換処理となるようにMix Column変換処理を実行するものであり、詳細は後述する。このことにより、第２Round Key加算回路２０８によるRound Key加算処理とMix Column変換回路２１０によるMix Column変換処理を実行した１２８ビットの演算結果が中間レジスタ兼Shift Row変換回路２０６に格納される。このようにして、第004サイクル～第012サイクルの９サイクルで１つのRound処理が実行される。

【００４４】以下、第２Round～第（Nr－１）Roundにおいて、第１Roundと同様の処理を実行する（ただし、Nrは最終Round処理を含むRound処理の繰り返し回数であり、表１に示すように、鍵長により異なる回数が規定されている）。

【００４５】第（Nr）Round（最終Round）では、Shift Row変換処理、Byte Sub変換処理、Round Key加算処理を実行する。このために、第（Nr＊９－５）サイクルにおいて、１２８ビットのShift Row変換処理を実行する。このとき、第２セレクタ２０３のセレクタ位置はいずれの位置であってもよいが、次のサイクルでの処理を考慮してｂの位置とすることが好ましい。

【００４６】次に、第（Nr＊９－４）サイクル～第（Nr＊９－１）サイクルにおいて、第２セレクタ２０３のセレクタ位置をｂとし、中間レジスタ兼Shift Row変換回

路２０６に格納されているデータを３２ビットずつシフトさせながら読み出してByteSub変換回路２０７に入力する。このとき、加算データセレクタ２０５により選択されるデータを"0"とすることで、第１Round Key加算回路２０４をマスク状態とする。Byte Sub変換回路２０７による演算結果は、３２ビット毎に中間レジスタ兼Shift Row変換回路２０６に順次格納される。このことにより、１２８ビットのByte Sub変換処理が行われ、演算結果が中間レジスタ兼Shift Row変換回路２０６に格納されることとなる。

【００４７】第（Nr＊９）サイクル～第（Nr＊９＋３）サイクルでは、Round Key加算処理を実行する。ここでは、第２セレクタ２０４のセレクタ位置をｄとすることで、中間レジスタ兼Shift Row変換回路２０６に格納されているデータを３２ビットずつシフトさせながら読み出して第２セレクタ２０３を介して第１Round Key加算回路２０４に入力する。このとき、加算データセレクタ２０５により選択されるデータを鍵スケジュール部から供給される拡大鍵とすることで、第１Round Key加算回路２０４による３２ビット毎のRound Key加算処理を実行することができる。第１Round Key加算回路２０４においてRound Key加算処理された演算結果は、３２ビット毎に中間レジスタ兼Shift Row変換回路２０６にそれぞれシフトされながら格納される。このことにより、第（Nr＊９）サイクル～第（Nr＊９＋３）サイクルで、Round Key加算処理を実行した１２８ビットの演算結果が中間レジスタ兼Shift Row変換回路２０６に格納される。このようにして、第（Nr＊９－５）サイクル～第（Nr＊９＋３）サイクルの９サイクルで最終Round処理が実行される。

〈中間値レジスタ兼Shift Row変換回路〉中間値レジスタ兼Shift Row変換回路の１実施例を図５に示す。

【００４８】ここでは、８ビット単位で処理を行うシフトレジスタを４個設けた構成となっている。１番目のシフトレジスタは、４つのフリップフロップ３０２，３０４，３０６，３０８が順次接続される構成でなり、各フリップフロップ３０２，３０４，３０６，３０８にはそれぞれ入力を選択するセレクタ３０１，３０３，３０５，３０７が接続されている。１番目のセレクタ３０１には、入力データIN0とフリップフロップ３０２の出力とが入力されておりいずれか一方をフリップフロップ３０２に入力するように構成されている。また、２～４番目のセレクタ３０３，３０５，３０７にはそれぞれ前段のフリップフロップ３０２，３０４，３０６の出力およびフリップフロップ３０４，３０６，３０８の出力とが入力されておりいずれか一方をフリップフロップ３０４，３０６，３０８に入力するように構成されている。

【００４９】２番目のシフトレジスタは、４つのフリップフロップ３１２，３１４，３１６，３１８が順次接続される構成でなり、各フリップフロップ３１２，３１４，３１６，３１８にはそれぞれ入力を選択するセレクタ３１１，３１３，３１５，３１７が接続されている。１番目のセレクタ３１１には、入力データIN1とフリップフロップ３１２の出力およびフリップフロップ３１８の出力が入力されておりいずれか一方をフリップフロップ３１２に入力するように構成されている。また、２～４番目のセレクタ３１３，３１５，３１７にはそれぞれ前段のフリップフロップ３１２，３１４，３１６の出力およびフリップフロップ３１４，３１６，３１８の出力とが入力されておりいずれか一方をフリップフロップ３１４，３１６，３１８に入力するように構成されている。

【００５０】３番目のシフトレジスタは、４つのフリップフロップ３２２，３２４，３２６，３２８が順次接続される構成でなり、各フリップフロップ３２２，３２４，３２６，３２８にはそれぞれ入力を選択するセレクタ３２１，３２３，３２５，３２７が接続されている。１番目のセレクタ３２１には、入力データIN2とフリップフロップ３２２の出力およびフリップフロップ３２６の出力が入力されておりいずれか一方をフリップフロップ３２２に入力するように構成されている。また、２番目のセレクタ３２３には前段のフリップフロップ３２２の出力、フリップフロップ３２４の出力およびフリップフロップ３２８の出力が入力されておりいずれか一方をフリップフロップ３２４に入力するように構成されている。３番目のセレクタ３２５には前段のフリップフロップ３２４の出力、フリップフロップ３２６の出力およびフリップフロップ３２２の出力が入力されておりいずれか一方をフリップフロップ３２６に入力するように構成されている。４番目のセレクタ３２７には前段のフリップフロップ３２６の出力、フリップフロップ３２８の出力およびフリップフロップ３２４の出力が入力されておりいずれか一方をフリップフロップ３２８に入力するように構成されている。

【００５１】４番目のシフトレジスタは、４つのフリップフロップ３３２，３３４，３３６，３３８が順次接続される構成でなり、各フリップフロップ３３２，３３４，３３６，３３８にはそれぞれ入力を選択するセレクタ３３１，３３３，３３５，３３７が接続されている。１番目のセレクタ３３１には、入力データIN3とフリップフロップ３３２の出力およびフリップフロップ３３４の出力が入力されておりいずれか一方をフリップフロップ３３２に入力するように構成されている。また、２番目のセレクタ３３３には前段のフリップフロップ３３２の出力、フリップフロップ３３４の出力およびフリップフロップ３３６の出力が入力されておりいずれか一方をフリップフロップ３３４に入力するように構成されている。３番目のセレクタ３３５には前段のフリップフロップ３３４の出力、フリップフロップ３３６の出力およびフリップフロップ３３８の出力が入力されておりいずれ

か一方をフリップフロップ３３６に入力するように構成されている。４番目のセレクタ３３７には前段のフリップフロップ３３６の出力、フリップフロップ３３８の出力およびフリップフロップ３３２の出力が入力されておりいずれか一方をフリップフロップ３３８に入力するように構成されている。

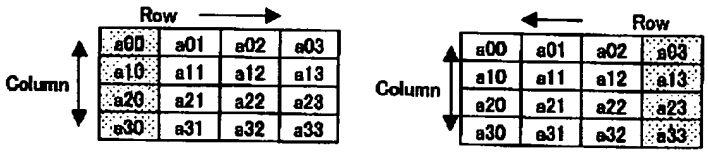
【００５２】このように構成した中間値レジスタ兼Shift Row変換回路を、各処理における中間値レジスタとして動作させる場合には、入力データIN0～IN3に各８ビット単位のデータを入力することで、各サイクルにおいて３２ビット単位で処理されたデータを格納することができる。さらに、各セレクタ３０１～３３７のセレクタ位置をｂにして、各フリップフロップのデータを後段にシフトさせつつ、入力データIN0～IN3にそれぞれ８ビット毎のデータを入力することで、４サイクルで１２８ビットのデータを入力することができる。１２８ビットのデータ入力が完了した場合、１サイクル目に入力された４つの８ビットデータは、それぞれフリップフロップ３０８，３１８，３２８，３３８にラッチされていることとなる。

【００５３】次にShift Row変換の動作について説明する。Rijndaelアルゴリズムでは、入力データをそれぞれ８bitのデータa00～a33に分割しこれをマトリクスとして処理を行い、暗号化と復号化でシフトの向きが逆になっている。本発明では、データの処理順をColumn（列）配列順とし、暗号化時と復号化時において逆順に処理することにより、同じ処理でShift Row変換処理を実現することができる。

【００５４】

【表４】

データ配列と処理順番



【００５５】表４左に示すように、各Row（行）におけるデータが左端のColumnから順に配列されているような場合には、暗号化時には、左端のColumnから順に処理が実行される。また、復号化時には、表４右に示すように、右端のColumnから順に処理が実行される。

【００５６】Shift Row変換処理では、暗号化時には、表４左に示されているような配列のデータ配列をRow毎に異なるシフト量でローテートシフトする。具体的には、表５に示すように、１行目をそのままとし、２行目を左１Byteローテートシフト、３行目を左２Byteローテートシフト、４行目を左３Byteローテートシフトする。このことにより、表５左に示す処理前の状態から表５右の処理後の状態となる。

【００５７】

【表５】

［暗号化］

処理前

a00	a01	a02	a03
a10	a11	a12	a13
a20	a21	a22	a23
a30	a31	a32	a33

左1Byteローテートシフト
左2Byteローテートシフト
左3Byteローテートシフト

処理後

a00	a01	a02	a03
a11	a12	a13	a10
a22	a23	a20	a21
a33	a30	a31	a32

【００５８】また、復号化時には、暗号化時と逆変換処理となるように、表４左に示されているような配列のデータ配列からRow毎に異なるシフト量でローテートシフトする。具体的には、表６に示すように、１行目をそのままとし、２行目を左３Byteローテートシフト、３行目を左２Byteローテートシフト、４行目を左１Byteローテートシフトする。このことにより、表６左に示す処理前の状態から表６右の処理後の状態となる。

【００５９】

【表６】

［復号化］

処理前

a00	a01	a02	a03
a10	a11	a12	a13
a20	a21	a22	a23
a30	a31	a32	a33

左3Byteローテートシフト
左2Byteローテートシフト
左1Byteローテートシフト

処理後

a00	a01	a02	a03
a13	a10	a11	a12
a22	a23	a20	a21
a31	a32	a33	a30

【００６０】本実施形態では、図５に示すような中間値レジスタ兼Shift Row変換回路を用いている。このことから、１２８ビットのデータの入力が完了した段階で、最初のサイクルで入力されたデータが最後段のフリップフロップ３０８，３１８，３２８，３３８にラッチされており、前段のフリップフロップに順にデータがラッチされている。データを取り出す場合には、１サイクル毎に右側に１Byteずつシフトしながら右端に位置する最後段のフリップフロップからデータが取り出されることとなる。したがって、データの処理順が右端側からであることを考慮してデータを配列し直すと、暗号化時のShift Row処理前は表７の左のような形態となる。

【００６１】

【表７】

［暗号化］

処理前					処理後			
a03	a02	a01	a00		a03	a02	a01	a00
a13	a12	a11	a10	右1Byteローテートシフト	a10	a13	a12	a11
a23	a22	a21	a20	右2Byteローテートシフト	a21	a20	a23	a22
a33	a32	a31	a30	右3Byteローテートシフト	a32	a31	a30	a33

【００６２】表５と同様のローテートシフトを行うためには、表７の右に示すように、１行目をそのままとし、２行目を右１Byteローテートシフト、３行目を右２Byteローテートシフト、４行目を右３Byteローテートシフトすることとなる。

【００６３】このような暗号化時におけるShift Row変換処理を行うためには、図５に示す中間値レジスタ兼Shift Row変換回路を用いて各セレクタを切換制御し、１２８ビット単位で一括してデータの入れ換えを行う。

【００６４】１番目のRowについては、シフトの必要がないため、各セレクタ３０１，３０３，３０５，３０７のセレクタ位置をａとする。２番目のRowについては、右１Byteローテートシフトを実行するために、セレクタ３１１のセレクタ位置をｃとし、それ以外のセレクタ３１３，３１５，３１７のセレクタ位置をｂとする。３番目のRowについては、右２Byteローテートシフトを実行するために、各セレクタ３２１，３２３，３２５，３２７のセレクタ位置をｃとする。４番目のRowについては、右に３byteローテートシフトを実行するために、各セレクタ３３１，３３３，３３５，３３７のセレクタ位置をｃとする。

【００６５】前述したようなShift Row変換処理を実行する前における中間値レジスタ兼Shift Row変換回路の各フリップフロップがラッチしている出力データを図５に示すようにそれぞれb00～b33とすると、各出力データは表８の右に示すような配列で各フリップフロップの出力にラッチされることとなる。

【００６６】

【表８】

ShiftRow変換動作モデル

ShiftRow変換前					ShiftRow変換後			
b03	b02	b01	b00	⇒	b03	b02	b01	b00
b13	b12	b11	b10		b10	b13	b12	b11
b23	b22	b21	b20		b21	b20	b23	b22
b33	b32	b33	b30		b32	b31	b30	b33

【００６７】復号化時には、表４のように右側の列から処理が実行されるため、表９左に示すように、データが配列される。

【００６８】

【表９】

［復号化］

処理前					処理後			
a00	a01	a02	a03		a00	a01	a02	a03
a10	a11	a12	a13	右1Byteローテートシフト	a13	a10	a11	a12
a20	a21	a22	a23	右2Byteローテートシフト	a22	a23	a20	a21
a30	a31	a32	a33	右3Byteローテートシフト	a31	a32	a33	a30

【００６９】表６と同様のローテートシフトを行うためには、表９の右に示すように、１行目をそのままとし、２行目を右１Byteローテートシフト、３行目を右２Byteローテートシフト、４行目を右３Byteローテートシフトすることとなる。

【００７０】したがって、前述した暗号化時におけるShift Row変換時と同様に、中間値レジスタ兼Shift Row変換回路の各セレクタのセレクタ位置を設定して、表８に示すような暗号化時におけるローテートシフト処理と全く同じ処理を実行することにより、復号化時におけるShift Row変換処理を実行することが可能となる。

【００７１】このことにより、同一の中間値レジスタ兼Shift Row変換回路で暗号化時および復号化時におけるShift Row変換処理を共用することができる。

〈Mix Column変換回路〉この実施形態で採用されるMix Column変換回路の一例を図６に示す。

【００７２】このMix Column変換回路は４つの第１演算器３５１，第２演算器３５２，第３演算器３５３，第４

演算器３５４を含んでいる。第１演算器３５１は、それぞれ８ビット単位で演算処理を実行する第１乗算器３６１，第２乗算器３６２，第３乗算器３６３，第４乗算器３６４と、各乗算器３６１～３６４の出力の排他的論理和演算を実行するEXOR回路３６５とで構成されている。その他の第２演算器３５２～第４演算器３５４についても、図示していないが、第１乗算器～第４乗算器の４つの乗算器とEXOR回路が設けられている。

【００７３】（a0j,a1j,a2j,a3j）で構成される第ｊ番目のColumnを（b0j,b1j,b2j,b3j）で構成されるColumnに変換処理を行うものとした場合、変換処理後の第ｊ番目のColumnのデータ（b0j,b1j,b2j,b3j）は、次のように表すことができる。

暗号化

b0j = 02*a0j + 03*a1j + 01*a2j + 01*a3j

b1j = 01*a0j + 02*a1j + 03*a2j + 01*a3j

b2j = 01*a0j + 01*a1j + 02*a2j + 03*a3j

b3j = 03*a0j + 01*a1j + 01*a2j + 02*a3j

復号化

b0j = 0E*a0j + 0B*a1j + 0D*a2j + 09*a3j

b1j = 09*a0j + 0E*a1j + 0B*a2j + 0D*a3j

b2j = 0D*a0j + 09*a1j + 0E*a2j + 0B*a3j

b3j = 0B*a0j + 0D*a1j + 09*a2j + 0E*a3j

ここで、各項に乗算されている係数はそれぞれ２桁の１６進数である。

【００７４】このようなMix Column変換処理を実行するために、Column単位の３２ビットデータをそれぞれ第１演算器３５１～第４演算器３５４に入力して、第１乗算器～第４乗算器による乗算およびEXOR回路による演算を行う。

【００７５】各演算器３５１～３５４に設けられる乗算器３６１～３６４は、暗号化および復号のいずれの場合にも共通に使用することができるように、暗号化時の係数および復号化時の係数を備えており、演算時に選択することが可能に構成されている。

【００７６】第１演算器３５１の第１乗算器３６１では、入力されるデータに対して0x02倍または0x0E倍のいずれかの乗算が可能となっている。第２乗算器３６２では、入力されるデータに対して0x03倍または0x0B倍のいずれかの乗算が可能となっている。第３乗算器３６３では、入力されるデータに対して0x01倍または0x0D倍のいずれかの乗算が可能となっている。第４乗算器３６４では、入力されるデータに対して0x01倍または0x09倍のいずれかの乗算が可能となっている。

【００７７】第２演算器３５２の第１乗算器では、入力されるデータに対して0x01倍または0x09倍のいずれかの乗算が可能となっている。第２乗算器では、入力されるデータに対して0x02倍または0x0E倍のいずれかの乗算が可能となっている。第３乗算器では、入力されるデータに対して0x03倍または0x0B倍のいずれかの乗算が可能となっている。第４乗算器では、入力されるデータに対して0x01倍または0x0D倍のいずれかの乗算が可能となっている。

【００７８】第３演算器３５３の第１乗算器では、入力されるデータに対して0x01倍または0x0D倍のいずれかの乗算が可能となっている。第２乗算器では、入力されるデータに対して0x01倍または0x09倍のいずれかの乗算が可能となっている。第３乗算器では、入力されるデータに対して0x02倍または0x0E倍のいずれかの乗算が可能となっている。第４乗算器では、入力されるデータに対して0x03倍または0x0B倍のいずれかの乗算が可能となっている。

【００７９】第４演算器３５４の第１乗算器では、入力されるデータに対して0x03倍または0x0B倍のいずれかの乗算が可能となっている。第２乗算器では、入力されるデータに対して0x01倍または0x0D倍のいずれかの乗算が可能となっている。第３乗算器では、入力されるデータに対して0x01倍または0x09倍のいずれかの乗算が可能となっている。第４乗算器では、入力されるデータに対して0x02倍または0x0E倍のいずれかの乗算が可能となっている。

【００８０】このような第１演算器３５１～第４演算器３５４の第１乗算器～第４乗算器の各係数を暗号化時および復号化時において変更することにより、暗号化時と復号化時における回路構成を共通に使用することが可能となる。

〈Mix Column変換回路の乗算器〉Mix Column変換回路中に含まれる乗算器の一例を図７に示す。

【００８１】入力される８ビットデータ（a7,a6,a5,a4,a3,a2,a1,a0）に対し、係数（b3,b2,b1,b0）の乗算を行うものとする。この場合、８ビットデータ（a7,a6,a5,a4,a3,a2,a1,a0）に係数の各ビットを乗算する部分積演算部３７５～３７８を設けている。さらに、係数の最上位ビットを乗算する部分積演算部３７５の演算結果に対して、部分積演算部３７６の演算結果を１ビットシフトさせて加算する加算器３７１、部分積演算部３７７の演算結果をさらに１ビットシフトさせて加算する加算器３７２、部分積演算部３７８の演算結果をさらに１ビットシフトさせて加算する加算器３７３を備えている。加算器３７３の演算結果と部分積演算部３７５、各加算器３７１～３７３でオーバーフローしたキャリーが入力され、除数により除算される除算器３７４が設けられている。

【００８２】このような構成により、係数（b3,b2,b1,b0）として、暗号化時における係数と復号化時における係数とを選択的にセットすることでMix Column変換処理を暗号化時と復号化時において共通に使用することができる。

【００８３】（b3,b2,b1,b0）としてセットされる係数は、前述したように、各乗算器について２つずつ設定さ

れている。各乗算器における係数の組み合わせは、（0x02,0x0E），（0x03,0x0B），（0x01,0x0D），（0x01,0x09）の４通りであり、これを下位４ビットで表示すると、（0010,1110），（0011,1011），（0001,1101），（0001,1001）となる。このような係数において共通するビットの演算は部分積の制御を行わず、異なるビットの演算は加算処理を制御することで、回路規模を小さくすることが可能である。

【００８４】たとえば、係数が（0x01,0x0D）の組み合わせの場合、２進数で表現すると（0001,1101）となり、上位２ビットの部分積の加算結果を下位２ビットの部分積に加算するか否かを制御することで、２つの係数を選択して乗算することが可能となる。係数（0x01,0x0D）の組み合わせの場合における回路構成の例を図８に示す。

【００８５】図８では、入力される８ビットデータ（a7,a6,a5,a4,a3,a2,a1,a0）に対し、１ビットシフトさせて加算処理を実行する第１加算器３８１が設けられている。第１加算器３８１の出力は、制御ロジック３８２を介して第２加算器３８３に入力されている。この第２加算器３８３は、係数の最下位ビットによる部分積演算結果を加算するものであって、入力される８ビットデータに対して３ビットシフトさせて加算処理を実行するように構成されている。

【００８６】加算器３８３の演算結果と第１加算器３８１および第２加算器３８３でオーバーフローしたキャリーが入力され、除数により除算される除算器３８４が設けられている。

【００８７】制御ロジック３８２は、係数が0x01である場合には、上位２ビットの演算結果である第１加算器３８１の出力を第２加算器３８３に出力しない。また、制御ロジック３８２は、係数が0x0Dである場合には、上位２ビットの演算結果である第１加算器３８１の出力を第２加算器３８３に出力するように構成される。

【００８８】ここでの乗算は既約多項式$M(x)=x^8+x^4+x^3+x+1$とするＧＦ（2^8）上での乗算であり、加算はＧＦ（２）上での加算であるため、排他的論理和演算で実現することが可能である。

【００８９】このように、２つの係数の異なるビットにおける部分積の加算処理を制御することにより回路規模を小さくすることができ、回路規模を削減することが可能となる。

〈鍵スケジュール部〉鍵スケジュール部の回路構成の一例を図９に示す。

【００９０】この鍵スケジュール部は、主に、拡大鍵生成ロジック部１０１、拡大鍵レジスタ１２０および鍵入力レジスタ１３１とから構成されている。鍵入力レジスタ１３１は、８個の３２ビットレジスタk0～k7から構成される２５６ビットのレジスタであり、レジスタk0から順に暗号鍵が３２ビット毎に格納される。暗号鍵が２５６ビットの場合にはレジスタk0～k7全てにデータが格納され、暗号鍵が１９２ビットの場合にはレジスタk0～k5にデータが格納され、暗号鍵が１２８ビットの場合にはレジスタk0～k3にデータが格納される。

【００９１】鍵入力レジスタ１３１には、各レジスタk0～k7のうちいずれか１つの値を選択的に出力するセレクタ１３２が接続されている。このセレクタ１３２は、鍵入力レジスタ１３１の２５６ビットデータから３２ビットのデータを選択して拡大鍵レジスタ１２０の最下位に入力するものである。

【００９２】拡大鍵レジスタ１２０は、３２ビット単位での処理が可能な８個のフリップフロップ１２１～１２８が順次接続されたシフトレジスタで構成されている。最下位に位置するフリップフロップ１２８には、セレクタ１３２の出力と拡大鍵生成ロジック部１０１の出力を選択するセレクタ１１３の出力が入力される。フリップフロップ１２８の出力W7KEYは、フリップフロップ１２７に入力されている。フリップフロップ１２７の出力W6KEYは、フリップフロップ１２６の前段にあるセレクタ１１２に入力されている。セレクタ１１２には、フリップフロップ１２７の出力W6KEYと拡大鍵生成ロジック１０１の出力とが入力されており、いずれか一方をフリップフロップ１２６に入力するように構成されている。

【００９３】フリップフロップ１２６の出力W5KEYは、フリップフロップ１２５に入力されている。フリップフロップ１２５の出力W4KEYは、フリップフロップ１２４の前段にあるセレクタ１１１に入力されている。セレクタ１１１には、フリップフロップ１２５の出力W4KEYと拡大鍵生成ロジック１０１の出力とが入力されており、いずれか一方をフリップフロップ１２４に入力するように構成されている。

【００９４】フリップフロップ１２４の出力W3KEYは、フリップフロップ１２３に入力されている。フリップフロップ１２３の出力W2KEYは、フリップフロップ１２２に入力されている。フリップフロップ１２２の出力W1KEYは、フリップフロップ１２１に入力されている。

【００９５】拡大鍵生成ロジック部１０１は、拡大鍵生成用定数Rconが格納されているＲＯＭ１０２、ＲＯＭ１０２から読み出される値と読出信号RCON#ENとの論理積演算を行うAND回路１０３、拡大鍵レジスタ１２０の最上位に位置するフリップフロップ１２１の出力W0KEYとAND回路１０３の出力が入力されて排他的論理和演算を実行するEXOR回路１０４を含んでいる。

【００９６】また、拡大鍵生成ロジック部１０１は、フリップフロップ１２１の出力W0KEY、フリップフロップ１２４の出力W3KEY、フリップフロップ１２６の出力W5KEY、フリップフロップ１２８の出力W7KEYが入力され、いずれか１つを選択的に出力するセレクタ１０５を備えている。セレクタ１０５の出力は、データのローテーション処理を行うRot Byte回路１０６、セレクタ１０７、

セレクタ１０９に入力されている。セレクタ１０７には、Rot Byte回路１０６の出力とセレクタ１０５の出力が入力されており、いずれかをSub Byte回路１０８に供給する。Sub Byte回路１０８は、３２ビット分のByte Sub変換処理を実行するものであり、その出力がセレクタ１０９に供給する。セレクタ１０９は、Sub Byte回路１０８の出力とセレクタ１０５の出力が入力され、いずれか一方を出力する。拡大鍵生成ロジック部１０１は、EXOR回路１０４の出力とセレクタ１０９の出力が入力され、排他的論理和演算を実行するEXOR回路１１０を備えている。

【００９７】このように構成される鍵スケジュール部では、　ラウンドファンクション部のRound Key加算処理に用いる拡大鍵の生成、　暗号化時における鍵入力レジスタの書き換えと、暗号化と復号化終了に伴う拡大鍵初期値のセットアップ、　復号化時の鍵入力レジスタの書き換えに伴う拡大鍵初期値のセットアップなどの機能を備える。

【００９８】ラウンドファンクション部のRound Key加算処理で用いられるRound Keyは、鍵長２５６ビットの場合では、Initial Round KeyとRound Key01～Round Key14の合計１５個必要となる。各Round keyは処理ブロック長に対応して１２８ビットで構成されており、鍵スケジュール部で生成される３２ビットの拡大鍵で各Round keyを充当するためには、W00～W59の６０個の拡大鍵が必要となる。このような拡大鍵W00～W59は、暗号化時にはW00→W59の順に使用され、復号化時にはW59→W00の順に使用される。この実施形態では、表１０に示すように、暗号化時にW00→W59の順に拡大鍵を生成するとともに、復号化時においてW59→W00の順に拡大鍵の生成を行うように構成する。

【００９９】

【表１０】

拡大鍵スケジュール（例：鍵長==256bit）

No.	暗号化	復号化	
00	W00=0x0)	W59	Initial Round Key
01	W01=(0x1)	W58	
02	W02=(0x2)	W57	
03	W03=(0x3)	W56	
04	W04=(0x4)	W55	
05	W05=(0x5)	W54	
06	W06=(0x6)	W53	Round Key01
07	W07=(0x7)	W52	
08	W08=W00^SubByte(RotByte(W07))^Rcon[1]	W51=W59^W58	
09	W09=W01^W08	W50=W58^W57	
10	W10=W02^W09	W49=W57^W56	Round Key02
11	W11=W03^W10	W48=W56^SubByte(RotByte(W55))^Rcon[7]	
12	W12=W04^SubByte(W11)	W47=W55^W54	
13	W13=W05^W12	W46=W54^W53	
14	W14=W06^W13	W45=W53^W52	Round Key03
15	W15=W07^W14	W44=W52^SubByte(W51)	
16	W16=W08^SubByte(RotByte(W15))^Rcon[2]	W43=W51^W50	
17	W17=W09^W16	W42=W50^W49	
18	W18=W10^W17	W41=W49^W48	Round Key04
19	W19=W11^W18	W40=W48^SubByte(RotByte(W47))^Rcon[8]	
20	W20=W12^SubByte(W19)	W39=W47^W46	
21	W21=W13^W20	W37=W46^W45	
22	W22=W14^W21	W38=W45^W44	Round Key05
23	W23=W15^W22	W35=W44^SubByte(W43)	
	省略		
52	W52=W44^SubByte(W51)	W07=W15^W14	
53	W53=W45^W52	W06=W14^W13	
54	W54=W46^W53	W05=W13^W12	Round Key13
55	W55=W47^W54	W04=W12^SubByte(W11)	
56	W56=W48^SubByte(RotByte(W55))^Rcon[7]	W03=W11^W10	
57	W57=W49^W56	W02=W10^W09	
58	W58=W50^W57	W01=W09^W08	Round Key14
59	W59=W51^W58	W00=W08^SubByte(RotByte(W07))^Rcon[1]	

【０１００】ここで、暗号化時の拡大鍵W08は、W08=W00^Sub Byte(Rot Byte(W07))^Rcon[1]の式により、W00とSub Byte(Rot Byte(W07))と定数Rcon[1]との排他的論理和で演算される。A^A=Aという性質を利用すると、拡大鍵W00は、W00= W08^Sub Byte(Rot Byte(W07))^Rcon[1]で表され、W08とW07から生成できることがわかる。よって、復号化では、一旦、W00=>W59まで生成して、暗号化の時と逆の順番W59=>W00で拡大鍵を生成することが可能である。これにより、復号化時もすべての拡大鍵をメモリに格納しておく必要はなく、各ラウンドで必要な拡大鍵だけを生成しながら、復号処理を行うことが可能となる。

【０１０１】まず、ラウンドファンクション部でのRound Key加算用拡大鍵の生成について説明する。表１０に示すように各ラウンドのRound Key加算で4個の拡大鍵(32bit単位)が使用され、拡大鍵はラウンドファンクションのMix Column変換 + Round Key加算のバックグラウンドで演算されるため4個の拡大鍵は4サイクルで作成すれ

ばよい。このため図９に示すような回路構成において、１サイクルで１個の拡大鍵を生成する構成となっている。拡大鍵レジスタ１２０はシフトレジスタで構成されており、ラウンドファンクション部で使用される現在の拡大鍵は、フリップフロップ１２１の出力W0KEYを用いる。

【０１０２】拡大鍵生成ロジック部１０１のセレクタ１０５(SEL#B)は、表１１に示すように、鍵長と暗号化/復号化の2種類の条件により切り換え制御が行われる。また、拡大鍵生成ロジック部１０１の出力が供給されるセレクタ１１１，１１２，１１３（SEL#E～SEL#G）は、鍵長に基づいて、表１２のように設定される。ただし、暗号鍵が初期値として入力される際には、各セレクタ１１１～１１３のセレクタ位置はｂが選択される。さらに、セレクタ１０７，１０９(SEL#C～D)は、表１３に示すように、各拡大鍵の生成論理により切り換え制御が行われる。ＲＯＭ１０２には、EXOR回路１０４に供給される定数Rcon[i]が格納されており、アドレスｉ番地に対応する各定数Rcon[i]が表１４に示すように格納されている。

【０１０３】

【表１１】

SEL_B制御

鍵長	暗号化	復号化
128bit	W3KEY	W1KEY
192bit	W5KEY	W1KEY
256bit	W7KEY	W1KEY

【０１０４】

【表１２】

SEL_E～SEL_G制御

鍵長	SEL_E	SEL_F	SEL_G
128bit	a	b	b
192bit	b	a	b
256bit	b	b	a

【０１０５】

【表１３】

SEL_C～SEL_D制御

	論理	SEL_C	SEL_D
拡大鍵	W[i]=W[i-Nk]^W[i-1]	*	b
	W[i]=W[i-Nk]^SubByte(W[i-1])	b	a
	W[i]=W[i-Nk]^SubByte(RotByte(W[i-1]))^Rcon[i/Nk]	a	a
	ByteSub	c	b

*:don't care

【０１０６】

【表１４】

Rcon ROMテーブル

Rcon_Addr	Hex	Bin
01	0x01	0000_0001
02	0x02	0000_0010
03	0x04	0000_0100
04	0x08	0000_1000
05	0x10	0001_0000
06	0x20	0010_0000
07	0x40	0100_0000
08	0x80	1000_0000
09	0x1B	0001_1011
10	0x36	0011_0110

【０１０７】表１０に示すように、鍵長が２５６ビットである場合の各演算について回路動作を説明する。なお、ラウンドファンクション動作前に、鍵入力レジスタ１３１の各レジスタk0～k7の値をロードすることによって、No.00～07からの初期値が拡大鍵レジスタ１２０の各フリップフロップ１２１～１２８にセットされているものとする。

【０１０８】暗号化時における拡大鍵W08は、表１０に示されているように、W08=W00^Sub Byte(Rot Byte(W07))^Rcon[1]で演算される。このW08=W00^Sub Byte(Rot Byte(W07))^Rcon[1]の演算開始時には、フリップフロップ１２１の出力W0KEYにはW00がセットされており、EXOR回路１０４に入力される。また、フリップフロップ１２８の出力W7KEYにはW07がセットされており、このW07がセレクタ１０５（SEL#B）に入力されている。

【０１０９】ROM１０２のRconアドレスを"1"にし、AND回路１０３に入力される読出信号RCON#ENをイネーブルにして、Rcon[1]^W00の演算をEXOR回路１０４で行い、その結果をEXOR回路１１０に入力する。一方、セレクタ１０５（SEL#B）を通過したW07は、Rot Byte回路１０６とSub Byte回路１０７による処理が行われ、Sub Byte(Rot Byte(W07))の演算結果がEXOR回路１１０に入力される。よって、EXOR回路１１０では、W08=W00^Sub Byte(Rot Byte(W07))^Rcon[1]の演算が行われる。

【０１１０】次に、拡大鍵W09=W01^W08の演算処理を説明する。W09=W01^W08の演算開始時には、フリップフロップ１２１の出力W0KEYにはW01がセットされており、EXOR回路１０４にこれが入力されている。フリップフロップ１２８の出力W7KEYにはW08がセットされており、セレクタ１０５（SEL#B）に入力されている。AND回路１０３に入力される読出信号RCON#ENをディセーブルにして、フリップフロップ１２１から入力されるW01がそのままEXOR回路１１０に入力されるように設定する。このとき、セレクタ１０９（SEL#D）ではセレクタ位置がｂに設定されており、セレクタ１０５（SEL#B）を通過したW08が、セレクタ１０９を介してそのままEXOR回路１１０に入力される。

【０１１１】よって、EXOR回路１１０では、W09=W01^W08の演算が行われる。W10～W11、W13～W15も同じパスで

それぞれ演算が行われる。拡大鍵W12の演算処理を説明する。拡大鍵W12=W04^SubByte(W11)で演算され、この演算の開始時には、フリップフロップ１２１の出力WOKEYにはW04がセットされ、EXOR回路１０４に入力されている。また、フリップフロップ１２８の出力W7KEYにはW11がセットされておりセレクタ１０５（SEL#B）に入力されている。AND回路１０３に入力される読出信号RCON#ENをディセーブルとして、W04がそのままEXOR回路１０４に入力されるように設定する。一方、セレクタ１０７（SEL#C）ではセレクタ位置がｂに設定されており、セレクタ１０５（SEL#B）を通過したW11が、セレクタ１０７（SEL#C）を介してSub Byte回路１０８に入力される。このことにより、Sub Byte回路１０８によるSub Byte処理が行われ、Sub Byte(W11）の演算結果がEXOR回路１１０に入力される。よって、EXOR回路１１０では、W12=W04^Sub Byte(W11)の演算が行われる。

【０１１２】以上の要領で全ての拡大鍵について演算が行われる。次に、暗号化時の鍵入力レジスタ１３１の書換えと、暗号化と復号化終了に伴う拡大鍵初期値のセットアップについて説明する。このセットアップ動作は、次の暗号化もしくは復号化に備え、鍵入力レジスタ１３１に格納されている拡大鍵初期値を拡大鍵レジスタ１２０に転送する動作である。

【０１１３】鍵入力レジスタ１３１にセットされた拡大鍵初期値は、セレクタ１３２(SEL#A)により３２ビット単位でデータ選択が行われ、セレクタ１１３(SEL#G)のセレクタ位置ｂを経由して拡大鍵レジスタ１２０にセットされる。拡大鍵レジスタ１２０は、前述したようにシフトレジスタ構成であり、フリップフロップ１２８（FF7）=>フリップフロップ１２７（FF6）=>フリップフロップ１２６（FF5）=>フリップフロップ１２５（FF4）=>フリップフロップ１２４（FF3）=>フリップフロップ１２３（FF2）=>フリップフロップ１２２（FF1）=>フリップフロップ１２１（FF0）へとシフトして行き、８サイクルで全ての拡大鍵初期値が転送される。なお、セレクタ１３２（SEL#A）で選択される鍵入力データは鍵入力レジスタ１３１のレジスタk0,k1,k2,k3,k4,k5,k6,k7の順番である。

【０１１４】復号化時の鍵入力レジスタ１３１の書換えに伴う拡大鍵初期値のセットアップについて説明する。表１０のように、復号化においては、拡大鍵初期値を暗号化での最終拡大鍵のセット(W59～W52)とする必要がある。鍵入力レジスタ１３１の書換えにより、鍵入力レジスタ１３１にセットされたデータを前述した方法で、一旦、拡大鍵レジスタ１２０に転送し、暗号化の拡大鍵生成論理に従い、最終拡大鍵のセット（W52～W59）まで図９の回路を動作させる。

【０１１５】この最終拡大鍵のセットの生成期間に、W52の生成時にW52を鍵入力レジスタ１３１のレジスタk7に転送し、W53の生成時にW53をレジスタk6に転送し、W54の生成時にW54をレジスタk5に転送し、W55の生成時にW55をレジスタk4に転送し、W56の生成時にW56をレジスタk3に転送し、W57の生成時にW57をレジスタk2に転送し、W58の生成時にW58をレジスタk1に転送し、W59の生成時にW59をk0に転送することにより鍵入力レジスタ１３１に逆の順番で最終拡大鍵をセットする。さらに、鍵入力レジスタ１３１の最終拡大鍵のセットを前述した方法で拡大鍵レジスタ１２０に転送することで復号化時の鍵入力レジスタの書換えに伴う拡大鍵初期値のセットアップが完了するこの後、セレクタ１０５（SEL#B）、セレクタ１０７（SEL#C）、セレクタ１０９（SEL#D）、セレクタ１１１～１１３（SEL#E～SEL#G）を表１１～表１３に示すようなセレクタ位置に設定し、復号化に必要な各拡大鍵を順次生成する。〈Byte Sub変換回路の共用化〉上述した鍵スケジュール部のSub Byte処理およびラウンドファンクション部のByte sub変換処理では、ともに３２ビット単位でのByte Sub変換処理を実行しているため、これらの処理回路を共用化することが考えられる。

【０１１６】たとえば、図９に示す鍵スケジュール部に設けられるSub Byte回路１０８をラウンドファンクション部のByte Sub変換回路と共用化することについて考察。図４に示すラウンドファンクション部における中間レジスタ兼Shift Row変換回路２０６からByte Sub変換回路２０７への入力BSINを、図９に示す拡大鍵生成ロジック部１０１のセレクタ１０７のセレクタ位置ｃに接続する。さらに、拡大鍵生成ロジック部１０１のSub Byte回路１０８からの出力を、図４のByte Sub変換回路２０７の出力BSOUTとしてセレクタ２０３に接続する。

【０１１７】Sub Byte回路１０８を用いてByte Sub変換処理を行う場合には、表１３に示すように、セレクタ１０７（SEL#C）のセレクタ位置をｃとし、セレクタ１０９（SEL#D）のセレクタ位置をｂとする。このことによって、拡大鍵生成ロジック部１０１のSub Byte回路１０８を用いてラウンドファンクション部のByte Sub変換処理を実行することが可能となる。

〈Byte Sub変換回路〉Byte Sub変換処理は、８ビット単位での逆数演算と行列演算との組み合わせで構成されており、暗号化時においては逆数演算を行った後に行列演算を実行し、復号化時においては行列演算を行った後に逆数演算を実行する。このようなByteSub変換処理を、暗号化時と復号化時において共通の回路で実施するために、図１０に示すような回路を提案する。

【０１１８】図１０に示すByte Sub変換回路３９１は、復号化用行列演算回路３９２と、セレクタ３９３、逆数演算回路３９４、暗号化用行列演算回路３９５、セレクタ３９６とで構成されている。

【０１１９】セレクタ３９３は、入力データと復号化用行列演算回路３９２の出力が入力されておりいずれか一方を逆数演算回路３９４に入力するように構成されている。また、セレクタ３９６は、逆数演算回路３９４の出

力と暗号化用行列演算回路３９５の出力が入力されており、いずれか一方を出力するように構成されている。

【０１２０】暗号化時には、セレクタ３９３を入力データ側にし、セレクタ３９６を暗号化用行列演算回路３９５側にする。また、復号化時には、セレクタ３９３を復号化用行列演算回路３９２側にし、セレクタ３９６を逆数演算回路３９４側にする。このことにより、暗号化時におけるByte Sub変換処理と、復号化時におけるByteSub変換処理とを共通の回路構成で実現することが可能となる。

【０１２１】暗号化時における行列演算処理は、次のような数１で表される。

【０１２２】

【数１】

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

【０１２３】これを展開すると、次に示す数２のように表すことができる。ただし、ここでの”＋”は排他的論理和演算で構成される。

【０１２４】

【数２】

$$\begin{aligned}
y_0 &= x_0 + x_4 + x_5 + x_6 + x_7 + 1 \\
y_1 &= x_0 + x_1 + x_5 + x_6 + x_7 + 1 \\
y_2 &= x_0 + x_1 + x_2 + x_6 + x_7 \\
y_3 &= x_0 + x_1 + x_2 + x_3 + x_7 \\
y_4 &= x_0 + x_1 + x_2 + x_3 + x_4 \\
y_5 &= x_1 + x_2 + x_3 + x_4 + x_5 + 1 \\
y_6 &= x_2 + x_3 + x_4 + x_5 + x_6 + 1 \\
y_7 &= x_3 + x_4 + x_5 + x_6 + x_7
\end{aligned}$$

【０１２５】また、復号化時における行列演算処理は、次のような数３で表される。

【０１２６】

【数３】

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 0&0&1&0&0&1&0&1 \\ 1&0&0&1&0&0&1&0 \\ 0&1&0&0&1&0&0&1 \\ 1&0&1&0&0&1&0&0 \\ 0&1&0&1&0&0&1&0 \\ 0&0&1&0&1&0&0&1 \\ 1&0&0&1&0&1&0&0 \\ 0&1&0&0&1&0&1&0 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

【０１２７】これを同様に展開すると、次に示す数４のように表すことができる。

【０１２８】

【数４】

$$\begin{aligned}
y_0 &= x_2 + x_5 + x_7 + 1 \\
y_1 &= x_0 + x_3 + x_6 \\
y_2 &= x_1 + x_4 + x_7 + 1 \\
y_3 &= x_0 + x_2 + x_5 \\
y_4 &= x_1 + x_3 + x_6 \\
y_5 &= x_2 + x_4 + x_7 \\
y_6 &= x_0 + x_3 + x_5 \\
y_7 &= x_1 + x_4 + x_6
\end{aligned}$$

【０１２９】暗号化用行列演算回路の一例を図１１に示す。ここでは、８ビットの入力レジスタ４０１、出力レジスタ４０３および排他的論理和と否定で構成される論理回路４０２で構成される。暗号化時において数２に示すような排他的論理和演算を実行するためには、論理回路４０２内の排他的論理和回路のうち、重複する演算処理を共用化することで１６個のEXORと４個のＮＯＴを組み合わせることにより実現することが可能となる。

【０１３０】復号化用行列演算回路の一例を図１２に示す。暗号化用行列演算回路と同様に、８ビットの入力レジスタ４０５、出力レジスタ４０７および排他的論理和と否定で構成される論理回路４０６で構成されている。ここでも暗号化用行列演算回路と同様に、復号化時において数４に示すような排他的論理和演算を実行するために、論理回路４０６内の排他的論理和回路のうち、重複する演算処理を共有化することで１３個のEXOR回路と２個のＮＯＴを組み合わせることで実現することが可能となる。

【０１３１】暗号化用行列演算回路の他の例を図１３に示す。この暗号化用行列演算回路は、入力レジスタ４１１、出力レジスタ４１４、定数保持用レジスタ４１３および排他的論理回路で構成される論理回路４１２を備え

ている。入力レジスタ４１１、出力レジスタ４１４、定数保持用レジスタ４１３は、いずれも８ビットのシフトレジスタで構成されており、クロックに同期して１ビットずつ右にローテートシフトする。

【０１３２】数１の右辺第１項の定数は、１つの行が３つの"0"と５つの"1"でなり、１つずつシフトするように構成されている。これから、入力レジスタ４１１のx0, x4, x5, x6, x7の各ビットをローテートシフトさせながら論理回路４１２に入力してEXOR演算を実行することで、数１の右辺第１項の行列演算を行うことができる。

【０１３３】また、定数保持用レジスタ４１３には、下位ビットから順に数１右辺第２項の定数がセットされている。定数保持用レジスタ４１３の値をローテートシフトさせながら最下位ビットの値を論理回路４１２に入力しEXOR演算を実行することで、数１右辺第２項の行列演算を行うことができる。

【０１３４】このことにより、入力レジスタ４１１にデータがセットされると、最初のクロックでy0が演算されて出力レジスタ４１４に格納される。次のクロックでは、y1が演算されて出力レジスタ４１４に格納され、以下順次演算処理を実行して８クロックで（y7, y6, y5, y4, y3, y2, y1, y0）の演算が完了する。この場合の論理回路４１２は５個のEXOR回路で数２の演算処理を実行することが可能となる。

【０１３５】これと同様に構成した復号化用行列演算回路の他の例を図１４に示す。この復号化用行列演算回路は、入力レジスタ４１５、出力レジスタ４１８、定数保持用レジスタ４１７および排他的論理回路で構成される論理回路４１６を備えている。入力レジスタ４１５、出力レジスタ４１８、定数保持用レジスタ４１７は、いずれも８ビットのシフトレジスタで構成されており、クロックに同期して１ビットずつ右にローテートシフトする。

【０１３６】数３の右辺第１項の定数は、１つの行が５つの"0"と３つの"1"の組み合わせでなり、１つずつシフトするように構成されている。これから、入力レジスタ４１５のx2, x5, x7の各ビットをローテートシフトさせながら論理回路４１６に入力してEXOR演算を実行することで、数３の右辺第１項の行列演算を行うことができる。

【０１３７】また、定数保持用レジスタ４１７には、下位ビットから順に数３右辺第２項の定数がセットされている。定数保持用レジスタ４１７の値をローテートシフトさせながら最下位ビットの値を論理回路４１６に入力しEXOR演算を実行することで、数３右辺第２項の行列演算を行うことができる。

【０１３８】このことにより、入力レジスタ４１５にデータがセットされると、最初のクロックでy0が演算されて出力レジスタ４１８に格納される。以下、暗号化時と同様に順次演算処理を実行して８クロックで（y7, y6, y5, y4, y3, y2, y1, y0）の演算が完了する。この場合の論理回路４１６は３個のEXOR回路で数４の演算処理を実行することが可能となる。

（付記１）暗号鍵から所定の処理ブロック長に対応するビット数の複数のRound Keyを生成し、入力データと前記Round Keyのうちの１つとEXOR演算を実行するEXOR演算部と、Byte Sub変換、Shift Row変換、Mix Column変換、Round Key加算を含むRound処理を複数回実行するRound処理部とを備えるラウンドファンクション部により入力データと前記Round Keyによる暗号化／復号化処理を前記処理ブロック長毎に実行する暗号回路であって、前記Round処理部が、入力データを前記処理ブロック長よりも小さい実行ブロック長に分割する第１セレクタと、前記実行ブロック長毎に前記Round Keyの値を入力データに加算する第１Round Key加算回路と、前記第１Round Key加算回路の出力を一時的に格納するとともに前記処理ブロック長によるShift Row変換を実行する中間レジスタ兼Shift Row変換回路と、前記中間レジスタ兼Shift Row変換回路の値が前記実行ブロック長毎に入力されByte Sub変換を実行するByte Sub変換回路と、前記中間レジスタ兼Shift Row変換回路の値が前記実行ブロック長毎に入力され前記Round Keyの値を前記実行ブロック長毎に加算する第２Round Key加算回路と、前記第２Round Key加算回路の出力に対してMix Column変換を実行するMix Column変換回路と、前記第１セレクタ、中間レジスタ兼Shift Row変換回路、Byte Sub変換回路、Mix Column変換回路の出力のうちいずれか１つを前記第２Round Key加算回路に出力する第２セレクタとを備えることを特徴とする暗号回路。

【０１３９】（付記２）前記実行ブロック長は８の倍数ビットである付記１に記載の暗号回路。

（付記３）前記処理ブロック長が１２８ビットであり、前記実行ブロック長が３２ビットである付記１または２に記載の暗号回路。

【０１４０】（付記４）前記暗号鍵の鍵長が１２８ビット、１９２ビット、２５６ビットのうちのいずれかである付記１～３に記載の暗号回路。

【０１４１】（付記５）前記Byte Sub変換回路は、入力データに対して行列演算を実行する復号化用行列演算部と、入力データと前記復号化用行列演算部の出力とのうちいずれかを出力する第３セレクタと、前記第３セレクタから出力されるデータに対して逆数演算を実行する逆数演算部と、前記逆数演算部から出力されるデータに対して行列演算を実行する暗号化用行列演算部と、前記逆数演算部の出力と前記暗号化用行列演算部の出力のうちいずれか一方を出力する第４セレクタとを備える付記１～４のいずれかに記載の暗号回路。

【０１４２】（付記６）前記復号化用行列演算部と前記暗号化行列演算部は、１クロックで８ビットの演算を行うようにEXOR回路を接続している付記５に記載の暗号回路。

【０１４３】（付記７）前記復号化行列演算部と前記暗号化行列演算部は、１クロックで１ビットの演算を行うようにEXOR回路を接続している付記５に記載の暗号回路。

【０１４４】（付記８）前記中間レジスタ兼Shift Row変換回路に入力されるデータのシフト量に関するシフトデータを暗号化時と復号化時において逆順に入力することにより、前記中間レジスタ兼Shift Row変換回路を暗号化と復号化に共通に使用する、付記１～７のいずれかに記載の暗号回路。

【０１４５】（付記９）前記Mix Column変換回路は、乗数固定の複数の乗算器と前記複数の乗算器の排他的論理和を演算するEXOR回路とを備え、各乗算器に入力されるデータと各乗算器に設定された乗数との間で行列演算を実行する、付記１～８のいずれかに記載の暗号回路。

【０１４６】（付記１０）前記Mix Column変換回路は、８ビット単位で演算が可能な４つの乗算器と前記４つの乗算器の出力に基づいて排他的論理和演算を実行するEXOR回路とを有する演算器を４つ備える付記９に記載の暗号回路。

【０１４７】（付記１１）前記乗算器が、２つの乗数を制御可能であり暗号化と復号化に共通に使用される付記９または１０に記載の暗号回路。

【０１４８】（付記１２）前記乗算器は、上位ビットからの加算値を制御するように構成される、付記１１に記載の暗号回路。

【０１４９】（付記１３）暗号鍵を前記実行ブロック長に応じたビット数に分割して出力する第５セレクタと、前記実行ブロック長毎にデータをラッチするフリップフロップ回路が複数段接続されたシフトレジスタと、前記シフトレジスタの最終段のフリップフロップ回路の出力と定数群のうちから選択される１つの定数との排他的論理和演算を実行する第１EXOR回路と、前記シフトレジスタのフリップフロップのうち暗号化時に演算対象となるフリップフロップの出力と復号化時に演算対象となるフリップフロップの出力とが入力されいずれか１つを選択的に出力する第６セレクタと、前記第６セレクタの出力をローテーション処理するRot Byte処理回路と、前記第６セレクタの出力と前記Rot Byte処理回路の出力とが入力され、いずれか１つを選択的に出力する第７セレクタと、前記第７セレクタの出力に対して前記実行ブロック長毎のByte Sub変換を実行するSub Byte処理回路と、前記第６セレクタの出力と前記Sub Byte処理回路の出力が入力され、いずれか１つを選択的に出力する第８セレクタと、前記第１EXOR回路の出力と前記第８セレクタの出力とに基づいて排他的論理和演算を実行する第２EXOR回路と、前記シフトレジスタのフリップフロップのうち暗号化時にその出力が演算対象となるフリップフロップに対して、前記第２EXOR回路の出力と、隣接する段のフリップフロップの出力とのうちいずれかを選択的に出力するシフトレジスタ部セレクタと、を備える、前記暗号鍵から所定の処理ブロック長に対応するビット数の複数のRound Keyを前記実行ブロック長に対応するビット数毎に分割された拡大鍵として生成する拡大鍵スケジュール回路を有する付記１～１２のいずれかに記載の暗号回路。

【０１５０】（付記１４）前記シフトレジスタは、３２ビット単位でデータ処理を実行する８個のフリップフロップ回路を備え、前記第６セレクタは、前記フリップフロップのうち下から２段目、４段目、６段目、８段目のフリップフロップの出力が入力され、いずれか１つを出力するように構成されている付記１３に記載の暗号回路。

【０１５１】（付記１５）前記第７セレクタに前記中間レジスタ兼Shift Row変換回路の出力が入力され、前記Sub Byte処理回路の出力を前記第２セレクタに入力することにより、前記Sub Byte処理回路と前記Round処理部のByte Sub変換回路とを共用する、付記１３または１４に記載の暗号回路。

【０１５２】

【発明の効果】本発明によれば、特定処理回路の処理データを所定の実行ブロック長に細分化することにより、ＡＥＳブロック暗号アルゴリズムをコンパクトな回路で実行可能となる。また、暗号化時における処理回路と復号化時における処理回路を共有化するとともに、鍵スケジュール部とラウンドファンクション部の回路の一部を共有化することでさらに回路規模を小さくすることが可能となる。、

【図面の簡単な説明】

【図１】RijndaelアルゴリズムによるＡＥＳの処理ブロック図。

【図２】鍵スケジュールのプログラムリスト。

【図３】想定される回路実装の例を示すブロック図。

【図４】本発明の１実施形態に採用されるラウンドファンクション部のブロック図。

【図５】中間レジスタ兼Shift Row変換回路のブロック図。

【図６】Mix Column変換回路のブロック図。

【図７】乗算器の構成を示すブロック図。

【図８】乗算器の他の構成を示すブロック図。

【図９】鍵スケジュール部のブロック図。

【図１０】Byte Sub変換回路のブロック図。

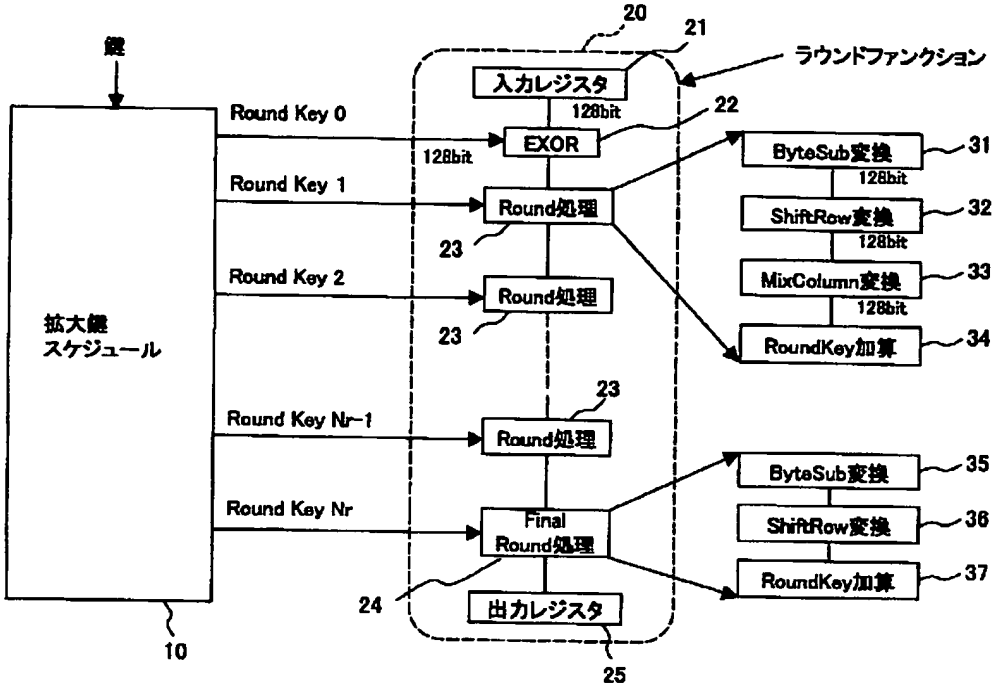
【図１１】暗号化用行列演算回路のブロック図。

【図１２】復号化用行列演算回路のブロック図。

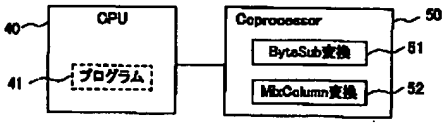
【図１３】暗号化用行列演算回路の他の例を示すブロック図。

【図１４】復号化用行列演算回路の他の例を示すブロック図。

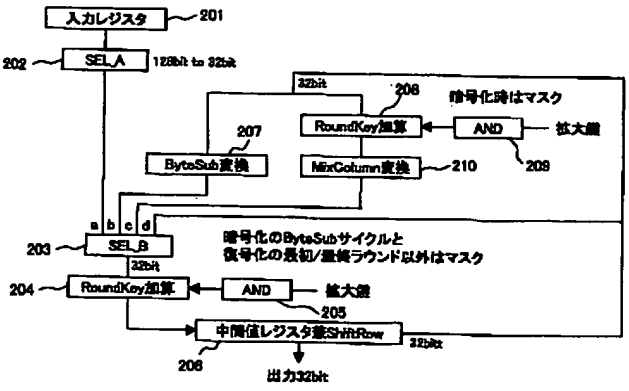
【図１】


鍵
拡大鍵
スケジュール
Round Key 0
Round Key 1
Round Key 2
Round Key Nr-1
Round Key Nr
10
20
21
ラウンドファンクション
入力レジスタ
128bit
22
EXOR
128bit
Round処理
23
Round処理
23
Round処理
23
Final
Round処理
24
出力レジスタ
25
ByteSub変換
128bit
31
ShiftRow変換
128bit
32
MixColumn変換
128bit
33
RoundKey加算
34
ByteSub変換
35
ShiftRow変換
36
RoundKey加算
37

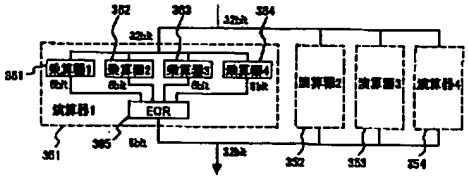
【図３】


CPU
40
プログラム
41
Coprocessor
50
ByteSub変換
51
MixColumn変換
52

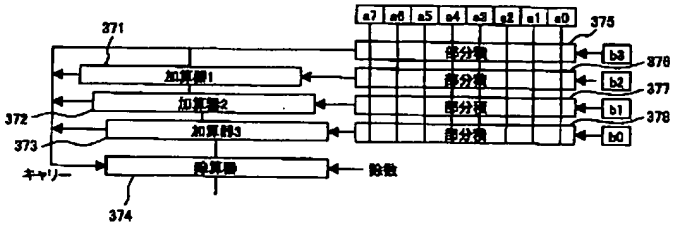
【図４】


入力レジスタ
201
SEL_A
202
128bit to 32bit
32bit
206
暗号化時はマスク
RoundKey加算
AND
拡大鍵
207
ByteSub変換
MixColumn変換
210
209
a b c d
SEL_B
203
暗号化のByteSubサイクルと
復号化の最初/最終ラウンド以外はマスク
32bit
204
RoundKey加算
AND
拡大鍵
205
中間値レジスタ兼ShiftRow
32bit
208
出力32bit

【図６】


352
353
32bit
354
32bit
演算器1
演算器2
演算器3
演算器4
8bit
8bit
8bit
8bit
351
演算器1
EOR
355
8bit
32bit
351
352
353
354

【図７】

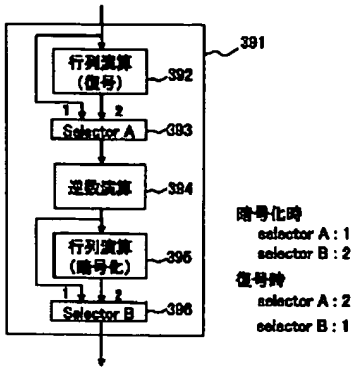

a7 a6 a5 a4 a3 a2 a1 a0
375
371
加算器1
加算器2
加算器3
372
373
キャリー
374
段数
b3
b2
b1
b0
376
377
378

【図２】

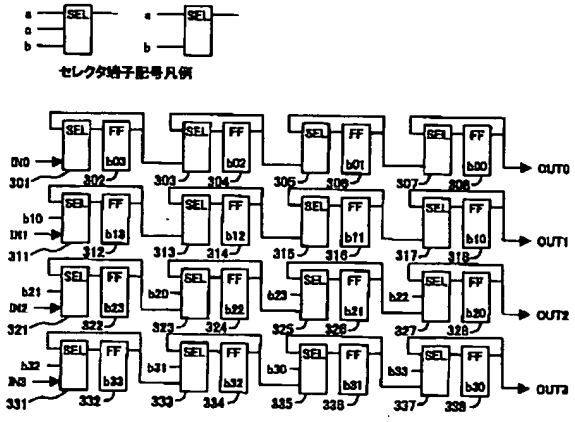
```
鍵長 = = 128bit or 192bit
KeyExpansion ( byte Key [ 4 * Nk]  word W [ Nb * ( Nr + 1 )]
{
        for ( i = 0 ; i < Nk ; i++ )
                W [ i ] = ( Key [ 4 * i ], Key [ 4 * i + 1 ], Key [ 4 * i + 3 ]);
        for ( i = Nk ; I < Nb * ( Nr + 1 ) ; i ++ )
        {
                temp = W [ i − 1 ];
                if ( i % Nk = = 0 )
                        temp = SubByte ( RotByte ( temp )) ^ Rcon [ i / Nk ];
                W [ i ] = W [ i − Nk ] ^ temp ;
        }
}


鍵長 = = 256bit
KeyExpansion ( byte Key [ 4 * Nk]  word W [ Nb * ( Nr + 1 )]
{
        for ( i = 0 ; i < Nk ; i++ )
                W [ i ] = ( Key [ 4 * i ], Key [ 4 * i + 1 ], Key [ 4 * i + 3 ]);
        for ( i = Nk ; i < Nb * ( Nr + 1 ) ; i ++ )
        {
                temp = W [ i − 1 ];
                if ( i % Nk = = 0 )
                        temp = SubByte ( RotByte ( temp )) ^ Rcon [ i / Nk ];
                else if ( i % Nk = = 4 )
                        temp = SubByte ( temp );
                W [ i ] = W [ i − Nk ] ^ temp ;
        }
}
```

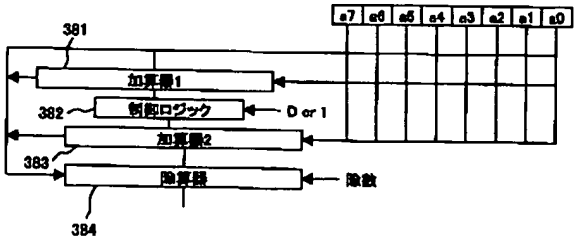
【図１０】



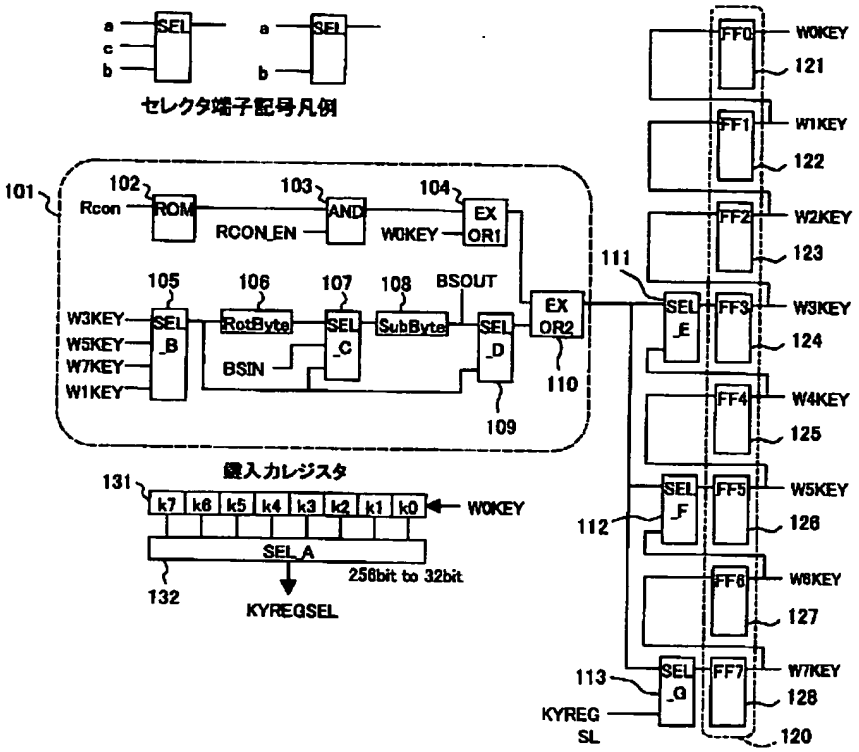
【図５】



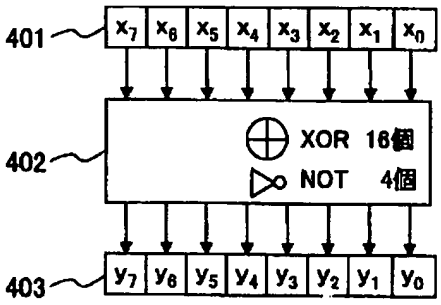
【図８】



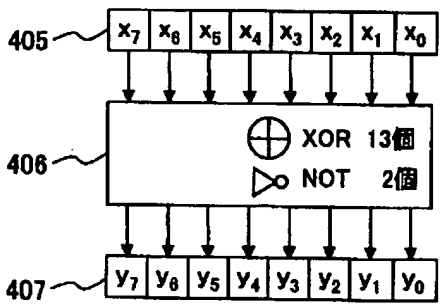
【図９】



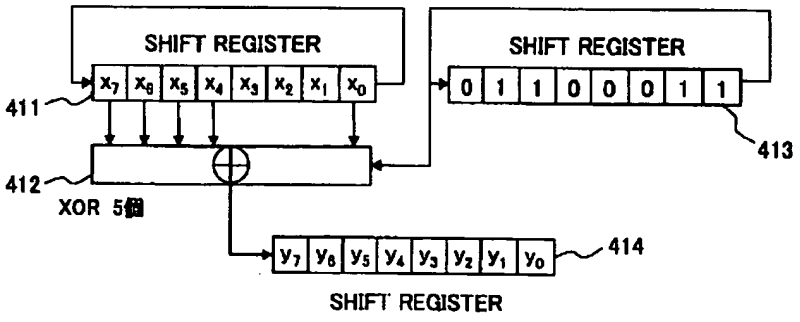
【図１１】



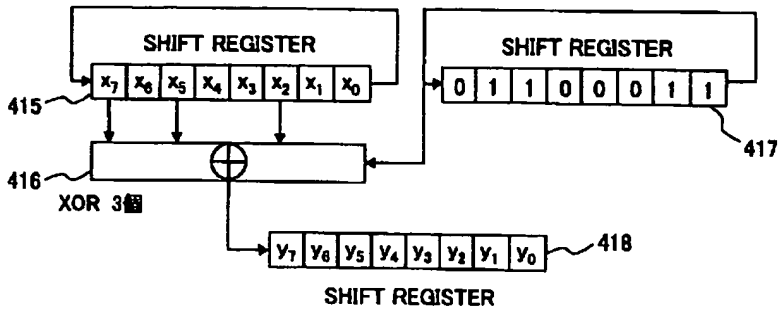
【図１２】



【図１３】



【図１４】



フロントページの続き

(72)発明者　林　朋弘
神奈川県横浜市港北区新横浜２丁目15番16号　株式会社富士通コンピュータテクノロジ内

(72)発明者　出口　千佳広
神奈川県横浜市港北区新横浜２丁目15番16号　株式会社富士通コンピュータテクノロジ内

(72)発明者　藤原　由実
神奈川県横浜市港北区新横浜２丁目15番16号　株式会社富士通コンピュータテクノロジ内

Ｆターム(参考)　5J104 AA18 JA03 NA02 NA22